



Espacenet

Bibliographic data: JP 5274275 (A)

No title available

Publication date: 1993-10-22

Inventor(s):

Applicant(s):

Classification: - **International:** G06F1/00; G06F15/16; G06F15/177; G06F21/22; G06F9/06; (IPC1-7): G06F15/16; G06F9/06
- **European:**

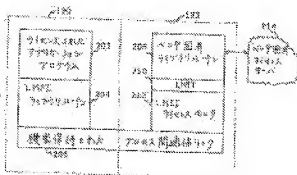
Application number: JP19920087680 19920311

Priority number (s): US19910667725 19910311

Also published as: ● JP 3270102 (B2)

Abstract of JP 5274275 (A)

PURPOSE: To simplify generation and maintenance of a part related to license servers in an application by unifying an interface for license servers when viewed from applications. **CONSTITUTION:** A licensed application program 202 which uses a service exists in a licensed application 120. When receiving a license request from this licensed application program 202, an operating system discriminates which one of plural license servers is used. The license request is converted to such format that one of plural license servers can accept it, and the converted license request is sent to one of plural license servers. When a response from one of plural license servers is received, this response is sent to the licensed application program 202.



特開平5-274275

(43)公開日 平成5年(1993)10月22日

(51)Int.Cl.³G 0 6 F 15/16
9/06

識別記号

3 7 0 N 9190-5L
4 5 0 Z 8944-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数10(全 14 頁)

(21)出願番号 特願平4-87680

(22)出願日 平成4年(1992)3月11日

(31)優先権主張番号 6 6 7, 7 2 5

(32)優先日 1991年3月11日

(33)優先権主張国 米国(US)

(71)出願人 590000400

ヒューレット・パッカード・カンパニー
アメリカ合衆国カリフォルニア州パロアル
ト ハノーバー・ストリート 3000

(72)発明者 ジェイムズ・ディー・ハンター

アメリカ合衆国コロラド州フォート・コリ
ンズ・レッドウィング・ロード2705

(72)発明者 ウィリアム・エイチ・ワトキンス

アメリカ合衆国コロラド州フォート・コリ
ンズ・ノーウッド・コート33200

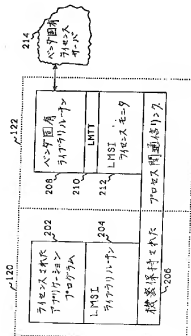
(74)代理人 弁理士 長谷川 次男

(54)【発明の名称】 ライセンス付与方法及びシステム

(57)【要約】

【目的】ネットワーク上でアプリケーションソフトウェアのライセンス付与管理(同時使用端末数に上限を与える、ライセンスを得ていないアプリケーションの起動の抑止等)を行うサーバとアプリケーションとの間のインターフェースを統一して、アプリケーションが各種のこれらサーバに対応するのを簡単にする。

【構成】起動されたアプリケーションからのライセンス要求はサーバに直接与えられずに、変換用のプロセスが先ず処理する。このプロセスは、現在使用されているサーバが何であるかを判定し、与えられた要求をこのサーバが受け入れる形式に変換する。



【特許請求の範囲】

【請求項1】下記のステップ(a)ないし(f)を有し、ライセンスされたアプリケーション・プログラムとオペレーティング・システム中の複数のライセンス・サーバとの間のインターフェースを取るための方法：

(a) 前記ライセンスされたアプリケーション・プログラムからライセンス要求を受け取る；

(b) 前記オペレーティング・システムが前記複数のライセンス・サーバのうちのどの1つを使用しているかを判定する；

(c) 前記ライセンス要求を前記複数のライセンス・サーバのうちの前記1つが受け入れることができるフォーマットに変換する；

(d) 前記変換されたライセンス要求を複数のライセンス・サーバのうちの前記1つに送る；

(e) 前記複数のライセンス・サーバのうちの前記1つから応答を受け取る；

(f) 前記応答を前記ライセンスされたアプリケーション・プログラムに送り出す。

【請求項2】下記のステップ(g)を更に有することを特徴とする請求項1記載の方法：

(g) 周期的なライセンス更新要求を前記複数のライセンス・サーバのうちの前記1つに自動的に送る。

【請求項3】前記ステップ(a)が下記のステップ(a1)及び(a2)を更に含むことを特徴とする請求項1記載の方法：

(a1) ライセンス・モニタが前記オペレーティング・システム中で現在アクティブであるかを判定する；

(a2) 前記ステップ(a1)において前記ライセンス・モニタがアクティブではないと判定された場合には、前記ライセンス・モニタをアクティブにする。

【請求項4】前記ステップ(e)が下記のステップ(e1)ないし(e3)を更に含むことを特徴とする請求項1記載の方法：

(e1) 前記複数のライセンス・サーバのうちの前記1つが、今ライセンスを得ることはできないことを指示した場合には、前記ライセンスされたアプリケーション・プログラムのオプション・要求を前記アプリケーション・プログラムのユーザに提示する；

(e2) 前記ユーザからの入力を得る；

(e3) 複数の動作のうちの前記ユーザからの前記入力によって判定された1つを実行する。

【請求項5】前記ステップ(e3)が下記のステップ(e3a)ないし(e3c)を更に含むことを特徴とする請求項4記載の方法：

(e3a) 前記ユーザからの入力が前記ライセンス要求を再度試みることを要求するものであった場合には、前記ライセンス要求を再試行する；

(e3b) 前記ユーザからの入力がライセンスが利用可能になるのを待つことを要求するものであった場合に

は、前記ライセンス要求を待ち行列に付ける；

(e3c) 前記ユーザからの要求が別の種類のライセンスを要求するものであった場合には、前記別の種類のライセンスの要求を送り出す。

【請求項6】前記ステップ(a)が下記のステップ(a1)を更に含むことを特徴とする請求項1記載の方法：

(a1) 前記オペレーティング・システムの別個のタスクで前記ライセンス要求をライセンス・モニタに送り出す。

【請求項7】前記ステップ(a1)は、前記ライセンス要求を前記ローカルライセンス・サーバに送り出す前に前記ライセンス要求をコード化するステップを更に含むことを特徴とする請求項6記載の方法。

【請求項8】下記の(a)ないし(g)を設け、ライセンスされたアプリケーション・プログラムとオペレーティング・システム中の複数のライセンス・サーバの間のインターフェースを取るシステム：

(a) 前記ライセンスされたアプリケーション・プログラムへのライセンス要求を受け取るライブラリ・モジュール手段；

(b) 前記オペレーティング・システムの別個のタスク中に置かれているライセンス・モニタ手段；

(c) 前記ライセンス・モニタ手段内にあり、前記複数のライセンス・サーバのうちのどの1つが前記オペレーティング・システムによって使用中であるかを判定する手段；

(d) 前記ライブラリ・モジュール手段からの前記ライセンス要求を前記ライセンス・モニタ手段に送り出す通信手段；

(e) 前記ライセンス要求を前記複数のライセンス・サーバのうちの前記1つが受け入れることができるフォーマットに変換する変換テーブル手段；

(f) 前記変換された要求を前記複数のライセンス・サーバのうちの前記1つに送り出す手段；

(g) 前記通信手段を使用して、前記複数のライセンス・サーバの内の1つから受け取った応答を前記ライブラリ・モジュール手段及び前記ライセンスされたアプリケーション・プログラムに送り出す応答手段。

【請求項9】前記ライセンス・モニタ手段は更に周期的なライセンス更新要求を自動的に前記複数のライセンス・サーバのうちの前記1つへ送り出す手段を設けたことを特徴とする請求項8記載のシステム。

【請求項10】下記の(a)ないし(f)を設け、ライセンスされたアプリケーション・プログラムとオペレーティング・システム中のライセンス・サーバの間でインターフェースを取る方法：

(a) 前記ライセンスされたアプリケーション・プログラムからライセンス要求を受け取る；

(b) 前記ライセンス要求を前記ライセンス・サーバが受け入れることのできるフォーマットに変換する；

(c) 前記交換された要求を前記ライセンス・サーバに送り出す；

(d) 前記ライセンス・サーバから応答を受け取る；

(e) 前記応答を前記ライセンスされたアプリケーション・プログラムに送り出す；

(f) 周期的なライセンス更新要求を前記ライセンス・サーバに自動的に送り出す。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はコンピュータシステムに関し、より詳細にはコンピュータシステム内でソフトウェアのライセンスを与えることに関する。更に詳細には本発明は共通のネットワークライセンスサーバを介してソフトウェアのライセンスを与えることに関する。

【0002】

【従来技術及びその問題点】 コンピュータソフトウェアへのライセンスの付与は、従来そのソフトウェアの使用をライセンスされたコンピュータ毎にそのソフトウェアのコピーを提供することによって行われてきた。このライセンスされたソフトウェアはそのコンピュータに「固定され」、他のコンピュータはそれを正当に使用することができなかった。複数のコンピュータがネットワークで結合されると、ソフトウェアを1つのネットワークノードに固定することは特に煩雑になる。この問題に対する解決策の1つとして、ネットワークライセンス管理プログラムまたはネットワークライセンスサーバと呼ばれるものを使用する方法がある。ネットワークライセンスサーバを使用すると、ソフトウェアはネットワークの任意のあるいはすべてのノードにインストールすることができるが、ある一定の数のライセンスだけをその供給業者から購入する。ライセンスサーバはソフトウェアの同時ユーザの数を購入したライセンスの数に制限する。これはユーザがそのソフトウェアの使用を開始する際にある形態のソフトウェアへのライセンス「トークン」を提供することによって行われる。すなわち、ユーザがソフトウェアの使用を希望するたびにソフトウェアはライセンストークンを受け取るネットワークメッセージをライセンスサーバに送る。トークンが提供できる場合はそれをソフトウェアに返し、かくしてこのソフトウェアはユーザのためにその機能を実行する。ユーザがソフトウェアの使用を終えると、トークンはライセンスサーバに返される。ライセンスがすべて使用中であるとき、次のユーザはトークンを受け取らず、トークンが利用可能となるのを待たねばならない。例えば、ネットワークが10のノードを有する場合、ソフトウェアはそれらのノードのすべてに設置することができる。購入されたライセンスがうつつかない場合、このソフトウェアを同時に使用できるユーザの数は5である。ネットワーク上の6番目のユーザがソフトウェアを使用しようとする場合は、ライセンスがないためこの6番目のユーザは待機しなけ

ればならない。この6番目のノード上のソフトウェアはライセンストークンを受け取るまで機能しない。このように、ソフトウェアの供給業者は最大数の同時ユーザに対して支払いを受けることができ、顧客は小規模な利用を開始して使用量が増えるにつれてライセンスを追加することができる。

【0003】 この方法は一般的になり、異なるソフトウェア供給業者から数種の異なるライセンスサーバが入手可能である。しかし、これには問題点がある。すなわち、顧客は利用可能なライセンスサーバのうちの任意のものを用いることができ、アプリケーションソフトウェアの供給業者はライセンスサーバソフトウェアの供給業者とは異なる場合があり、このアプリケーションソフトウェアの供給業者が異なるライセンスサーバのそれぞれにインターフェースを取るための異なるソフトウェアを提供しなければならない。さまざまなライセンスサーバとインターフェースを取るにはアプリケーションソフトウェアの供給業者は、各ライセンスサーバに1つずつ、合計で数種のバージョンのアプリケーションソフトウェアを開発しなければならない。これは大きな費用がかかる、ユーザに間違ったバージョンが送られた場合には相当な遅延を引き起こす。

【0004】 もう1つの問題点は、それぞれのライセンスサーバが異なるインターフェース条件を有することである。このため開発費が増し、またエラーの可能性が高くなる。もう1つ別の問題点として、新しいライセンスサーバが開発されると、既存のサーバの新しいバージョンが出るたびにソフトウェアアプリケーションを変更しなければならないという点である。

【0005】 ライセンスが得られないときに一貫したユーザインターフェースを提供することも問題の1つである。ライセンスが得られないときユーザに通知が行われねばならず、またアプリケーションを中止するかライセンス待ちの行列に入るかどうかの選択ができなければならない。このユーザインターフェースは各アプリケーション毎に、ライセンスサーバ1つずつについて書かなければならず、従ってユーザはそのすべてが同じ問題を決済するのに用いられる多数の異なるインターフェースを与えられる。

【0006】 当該技術分野において、アプリケーションソフトウェアがさまざまなライセンスサーバにインターフェースを取れるようにする共通インターフェースが必要とされている。当該技術分野において、かかる共通インターフェースであって更にアプリケーションソフトウェアが新しく開発されたライセンスサーバに簡単に適応できるようにするものが必要とされている。更に、ライセンス取得に問題があることをユーザに通知するための共通ユーザインターフェースが必要とされている。本発明はこれらの要望及びその他の要望に応えるものである。

【0007】

【目的】本発明の1つの目的は、共通ネットワークライセンスサーバソフトウェアを介してソフトウェアアプリケーションプログラムをライセンスする方法を提供することである。

【0008】本発明の別の目的は、かかるアプリケーションプログラムと複数のネットワークライセンスサーバの間に共通インターフェースを提供することである。

【0009】本発明の更に別の目的は、このようなライセンスの付与が、アプリケーションプログラム中の一点からの一組のプログラム呼び出しによってできるようにすることである。

【0010】本発明の更に別の目的は、ソフトウェアアプリケーションに対するライセンスが得られないとき、ユーザが利用可能な選択肢を意図するための単一のユーザインターフェースを提供することである。

【0011】

【概要】本発明の上述の目的及びその他の目的は、さまざまなライセンスサーバに共通のアプリケーションプログラムソフトウェアを提供するシステムにおいて達成される。このシステムは一組のプログラム呼び出しを提供し、この一組の呼び出しを各ライセンスサーバ用の別の呼び出しの組に変換する。この変換は変換テーブルを用いて行われ、この変換テーブルは新しく開発されたライセンスサーバあるいは既存のサーバの新バージョンにインターフェースするために簡単に更新することができる。

【0012】ほとんどのライセンスサーバの必要条件の1つに、アプリケーションプログラムがライセンスサーバにライセンス更新メッセージを定期的に送ってサーバにそのライセンスが依然として使用中であることを通知することがある。アプリケーションプログラムがこのメッセージを送らなると、ライセンスサーバはそれを時間切れ処理して、そのライセンスを別のユーザに与えてしまう。本発明のシステムはそのオペレーティングシステム環境内で別個のプロセスとして動作して、このアプリケーションプログラムを監視し、アプリケーションプログラムがユーザにサービスを提供し続けるかぎり、本発明のシステムは定期的なライセンス更新メッセージをライセンスサーバに送る。これによってアプリケーションプログラムが定期的にメッセージを与える作業から解放され、アプリケーションプログラムは初期化呼び出しを与えるだけでよく、アプリケーションプログラムがかなり簡略化される。

【0013】このシステムはまた、アプリケーションプログラムがライセンスを取得することができないとき、これをユーザに通知する。本発明のシステムがこの通知を実行するため、本発明のシステムを用いるすべてのソフトウェアアプリケーションに渡ってユーザインターフェースが一貫性を持ったものになる。

【0014】

【実施例】以下の説明は本発明の現在のところ考えられる最も好適な実施態様に関するものである。この説明は限定的な意味に解すべきではなく、本発明の一般の原理を説明する目的のためにのみ行うものである。本発明の範囲は特許請求の範囲を参照して判断すべきものである。

【0015】図1は本発明を採用したコンピュータシステムのブロック図を示す。図1において、コンピュータシステム100は処理要素102を含む。処理要素102はシステムバス104を介してコンピュータシステムの他の要素と通信する。キーボード106を使ってシステムのユーザがテキスト情報を入力でき、またマウス10によりユーザが図形情報を入力することができる。図形表示装置108はシステムがテキストと図形をユーザが見ることができるよう出力することを可能にする。ディスク112は本発明のソフトウェアとデータ、更にコンピュータシステム100内のソフトウェアアプリケーションのソフトウェアとデータをストアするのに用いられる。通信インターフェース114は、本発明においてコンピュータ100が取り付けられたネットワークの別のノード中のライセンスモジュールのリモートサーバとの通信に用いられる。メモリ116はUnixマルチタスクオペレーティングシステム118を含む。本発明の以下で説明する実施例においてはUnixオペレーティングシステムを用いるが、他のマルチタスクオペレーティングシステム、更にはMS-DOS等のシングルタスクオペレーティングシステムを本発明に用いることもできる。ライセンスされたアプリケーション120は、コンパイルされてこのライセンスされたアプリケーション120にリンクされたライブラリチェーンを介して本発明のサービスを使用する。これらのルーチンはライセンスモニタのローカル側サーバタスク122と通信し、このライセンスモニタのローカル側サーバタスク122は通信インターフェース114を介して供給業者に固有のライセンスサーバ(図1には示さない)と通信する。

【0016】図2は本発明の諸プロセスとこれらの相互作用を示す図である。図2において、図1のライセンスされたアプリケーション120を被検で示す。ライセンスされたアプリケーション120中には本発明のサービスを使用するライセンスされたアプリケーションプログラム202がある。ライセンスされたアプリケーション120にはライセンスされたアプリケーションプログラム202との通信のための一組のライセンス機構標準インターフェース(Licensed Mechanism Standard Interface, LMSI)ライブラリルーチン204がリンクされている。これらのルーチンについては以下に詳細に説明する。LMSIライブラリルーチン204は機構保護されたプロセス間通信リンク206上でライセンスモニ

タのローカル側サーバ122と通信する。リンク206はデータをコード化することによって機密保護される。当該技術分野ではデータをコード化するさまざまな手段が知られている。このコード化は、にせのライセンスモニタのローカル側サーバを供給することによってこのライセンスされたアプリケーションプログラムが不法に使用されるのを防止するために必要である。

【0017】ライセンスモニタのローカル側サーバ122はLMSIライブラリルーチン204からの通信を受けるLMSIライセンスモニタ212を含む。LMSIライセンスモニタ212は、ライセンスモニタ変換テーブル(License Monitor Translation Table、LMTT)210を用いて、ライセンスされたアプリケーションプログラム202からの要求を、特定のライセンスサーバの供給業者固有の特定の組のライブラリルーチン208への要求に変換する。すると、この供給業者固有のライブラリルーチン208は、普通は、ネットワークの別のノード内に配置されている供給業者固有のライセンスサーバ214に連絡する。この供給業者固有のライセンスサーバ214は、例えば本願出願人のNetLSライセンスサーバ、Highland SoftwareのFlexlmライセンスサーバ、Sun Microsystems Inc.のSunNETライセンスサーバといったいくつかの供給業者用のいくつかのルーチンの組のうちの1つとすることができ、

【0018】図3にはLMSIライブラリルーチン204(図2)のうちの1つのルーチンである、ライセンスモニタを初期化するためにライセンスされたアプリケーションプログラム202によって呼び出されるルーチンのフローチャートを示す。図3において、このルーチンに入った後、ブロック302でライセンスモニタのローカル側サーバ122が現在デーモン(daemon)として動作中であるかどうか判定される。Unixオペレーティングシステムでは、デーモンとは通常はオペレーティングシステム中の初期化中に起動されるオペレーティングシステム中の独立したプロセスである。この種のプロセスはアプリケーションプログラムに対するさまざまなサービスを行う。例えば、Unixでは印字はデーモンプロセスによって行われる。本発明のシステムにおいては、ライセンスモニタのローカル側サーバはデーモンとして実現することができる。あるいはライセンスされたアプリケーションを含むプロセスによって起動されるチャイルドプロセスとして実現することができる。ブロック302ではサーバデーモンが活動状態であるかどうか判定され、活動中でない場合、制御はブロック302にブロック304に移り、ライセンスモニタのローカル側サーバが、チャイルドプロセスとして起動される。チャイルドプロセスの起動後、あるいはデーモンがすでに活動中である場合、制御はブロック306に移り、ライセンスモニタのローカル側サーバへの機密保護された通

信が確立される。この機密保護された通信はライセンスモニタのローカル側サーバとライセンスされたアプリケーション120中のLMSIライブラリルーチン204との間のすべての伝送をコード化することによって作り出される。次にブロック308で通信が確立されたかどうか判定され、確立されていなければ制御はブロック308からブロック316に移り、ライセンスされたアプリケーションプログラム202にエラー表示がでる。機密保護された通信が確立されていれば、制御はブロック308からブロック310に移り、そこでLMSIライセンスモニタ212が呼び出されてLMTT210から供給業者固有の情報ハンドルが得られる。この「ハンドル」はアプリケーションプログラム202がLMSIライブラリルーチンへの他のすべての呼び出しの際に使用しなければならない情報へのポイントである。ブロック312ではこのハンドルが得られたかどうか判定され、得られていなければ制御はブロック316に移ってエラーが返される。ハンドルが得られていれば、制御はブロック312からブロック314に移り、ハンドルがライセンスされたアプリケーションプログラム202に返される。

【0019】図4はライセンスされたアプリケーションプログラム202が初期化の後にライセンスモニタのローカル側サーバと対話するために呼び出すルーチンのフローチャートを示す。このルーチンはライセンスの要求、ライセンスの返却、このアプリケーションのためのライセンスを使用している現在のユーザのリストの獲得、あるいはオプションとしてライセンスの現在の状態のチェック、等を行うために呼び出される。図4において、このルーチンへ入った後、ブロック402においてライセンスモニタのローカル側サーバが活動状態であるかどうか判定され、活動状態でなければ制御はブロック402からブロック408に移り、エラー状態が返される。サーバが活動状態である場合、すなわち、このルーチンを呼び出す前に図3の初期化ルーチンが呼び出されていた場合、制御はブロック402からブロック404に移り、アプリケーションプログラム202のための要求の実行に必要なパラメータがライセンスモニタのローカル側サーバ122に返される。次にブロック404はライセンスモニタのローカル側サーバ11からメッセージが返ってくるのを待ち、ブロック406でこのメッセージ中で返された状態をチェックする。この状態が、要求が実行されなかったことを示している場合、制御はブロック406からブロック408に移り、エラー表示が返される。あるいはブロック406は成功表示とともにライセンスされたアプリケーションプログラム202に返す。その要求によって得られたあらゆるデータを返す。

【0020】図5、図6及び図7はLMSIライセンスモニタ212(図2)のフローチャートを示す。このモ

ニタには、ライセンスモニタのローカル側サーバ機能を実行させるため、図3及び図4のルーチンからメッセージが送られる。図5において、このルーチンへ入った後、ブロック502でLMS I ライブラリルーチンから送られたメッセージが初期化のためのものであるかどうか判定される。メッセージが初期化のためのものである場合、制御はブロック502からブロック504に移り、ライセンスモニタのローカル側サーバが前に初期化されていたかどうか判定される。前に初期化されていなかった場合、制御はブロック504からブロック506に移り、オペレーティングシステムが図7のルーチンを周期的に呼び出すようにするためにインターバルタイマシグナルを設定する。この呼び出しは、ネットワークの別のノードに位置するライセンスモニタのリモート側サーバに周期的なライセンス更新要求を送って、このライセンスをこのライセンスされたアプリケーションプログラムが使用できる状態に維持するために用いられる。この周期的呼び出しが行われない場合、供給業者固有のライセンスサーバはライセンスされたアプリケーションプログラムを含むノードにはもはや機能していないものと見なして、ライセンスの使用を他にライセンスしてしまいう可能性はある。インターバルタイマシグナルの設定後、制御はブロック506からブロック508に移り、供給業者固有のライブラリルーチン208(図2)が呼び出されて初期化が行われる。本実施例に示すシステムとインターフェースする供給業者固有のライブラリルーチンのさまざまな組はそれぞれがなんらかの形態の初期設定を有する。ブロック508において特定の供給業者に許容される特定の形態を用いて供給業者固有のライブラリルーチンが呼び出される。初期化の後、ブロック510でこの初期化が成功であったかどうか判定され、成功でなければ制御はブロック524に移り、そこでエラー表示を返す。初期化が成功であった場合、あるいはライセンスモニタのローカル側サーバが前に初期化されていた場合、制御はブロック512に移り、ライセンスモニタ変換テーブル210(図2)中に、この特定の供給業者用の領域へのポインタを設定する。次にブロック514でこのポインタをLMS I ライブラリルーチン204へのハンドルとして返す。メッセージが初期化のためのものではない場合、制御はブロック502からブロック516に移り、この呼び出しがライセンスを要求するものであるかどうか判定される。この呼び出しがライセンス要求のためのものである場合、すなわち、この呼び出しがライセンス返却のため、あるいはアプリケーション用にライセンスを使用中である現在のユーザの数を得るためのものである場合、あるいはそれが現在のライセンスの状態をチェックするための呼び出しである場合、制御はブロック518に移り、供給業者固有のライブラリルーチン208(図2)が呼び出され、LMS I ライブラリルーチン204によって要求された情報を得るためにバ

ラメータが渡される。次にブロック520で呼び出しが成功であったかどうか判定され、成功でなかった場合、制御はブロック520からブロック524に移り、LMS I ライブラリルーチン204にエラーメッセージが返される。この呼び出しが成功であった場合、制御はブロック520からブロック522に移り、アプリケーションによって要求された情報が返される。

【0021】LMS I ライブラリルーチン204の送ったメッセージがライセンスを要求するものである場合、制御はブロック516から図6のブロック602に移る。ブロック602では供給業者固有のライブラリルーチン208を呼び出し、ライセンスを要求し、供給業者固有のライブラリルーチンが供給業者固有のライセンスサーバに連絡を取ってライセンストークンを得るのを待つ。呼び出しから戻った後、ブロック604でライセンスが得られたかどうか判定される。ライセンスが得られた場合、制御はブロック604からブロック624に移り、ライセンステーブルにこのライセンスのエントリを入れる。これは図7の周期的ライセンス更新呼び出しに用いられる。ブロック624は次にLMS I ライブラリルーチン204に成功を表す情報を返す。

【0022】ライセンスが得られなかった場合、制御はブロック604からブロック606に移り、ユーザに入力要求するユーザ要求が表示される。このユーザ要求の図9に示す。図9に示す表示はウィンドウシステムからの典型的な表示であり、ユーザがその選択を表示するのに用いることのできるいくつかの「ボタン」を提供する。アプリケーションプログラムはLMS I ライブラリルーチン204を呼び出すときこれらのボタンのどれでもディセーブルすることができる。これによってこのアプリケーションがシステムを完全に制御することが可能になる。ユーザは、ボタン902(再試行)を使って再試行を要求することができ、ボタン904(待ち行列)を使ってその要求を待ち行列に置くことができ、ボタン906(デモ)を使ってソフトウェアのデモンストレーションコピーを要求することができ、ボタン908(終了)を使ってライセンスを得る試みを中止できる。ステータスボックス910はブロック606が現在の状態を表示するメッセージを表示するのに用いられる。ステータスボックス910に示すメッセージ例「現時点ではこのアプリケーションに対するライセンスを得ることができません。ノードにロックされた有効なライセンスは発見できなかったし、応答したライセンスサーバにはこのアプリケーションに対するライセンスがインストールされていません。一あなたのシステム管理者に連絡してください」は、ライセンスされたアプリケーションプログラムが供給業者固有のライセンスサーバにとって未知のものであるときに発生する。

【0023】システムにウィンドウシステムが用いられていないとき、文字ベースのユーザインターフェースが

用いられる。このインターフェース（図示せず）は図9の表示と同様であるが、マウスの代わりに矢印キーとエンターキーを用いる。

【0024】ユーザ要求の表示後、ユーザ入力を得るために制御はブロック606からブロック608に移る。次にブロック610でユーザが再試行ボタン902上でマウスをクリックしたかどうか判定され、そうであった場合は制御はブロック610からブロック602に移り、ライセンス獲得を再度試みる。ユーザ入力が再試行のためのものでない場合、制御はブロック610からブロック612に移り、ユーザがマウスをデモボタン906上でクリックしたかどうか判定される。そうであった場合は制御はブロック612からブロック614に移り、アプリケーションにデモンストレーションが要求されていることを知らせて戻る。ユーザがデモボタンをクリックしていなかった場合、制御はブロック612からブロック615に移り、ユーザが終了ボタン908をクリックしたかどうか判定される。ユーザが終了ボタン908をクリックしていた場合は制御はブロック615からブロック619に移り、終了が要求されたことをアプリケーションに知らせて戻る。ユーザが終了ボタン908をクリックしていなかった場合は制御はブロック615からブロック616に移り、ユーザが待ち行列ボタン904をクリックしたかどうか判定される。ユーザが待ち行列ボタン904をクリックしていなかった場合は制御はブロック616からブロック606に戻り、ユーザ要求を再表示する。ユーザが待ち行列ボタン904をクリックした場合は制御はブロック616からブロック618に移り、そこで待ち行列要求が作成される。次にブロック620で供給業者固有ライブラリルーチン208を呼び出して、待ち行列要求を供給業者固有のライセンスサーバに送ってこのライセンスされたアプリケーションプログラムを待ち行列に入れて利用可能な次のライセンスを取得しようとする。

【0025】図7はタイムシグナルハンドルーチンのフローチャートを示す。このシグナルハンドラへの呼び出しはブロック506（図5）によってセットアップされる。上述したように、このルーチンは供給業者固有のライセンスサーバに周期的ライセンス更新要求を送ってこのライセンスをアクティブな状態に維持するのに用いられる。図7において、このルーチンへ入った後、ブロック702でライセンステーブルの先頭のあるいは次のエントリを得る。このテーブルはブロック624（図6）によってセットアップされる。ブロック704でテーブルの最後に達したかどうか判定され、最後まで来ている場合、ブロック704はタイムアウトから復帰する。ライセンステーブル中にあるいはそれ以上のエントリがまたある場合、制御はブロック704からブロック706に移り、このブロック706で供給業者固有ライブラリルーチン208を呼び出し、ライセンスをア

クティブな状態に維持するために、このライブラリルーチンにネットワーク上に供給業者固有のライセンスサーバへのメッセージを送らせる。ライセンスが依然としてアクティブな状態である場合、供給業者固有のライセンスサーバは供給業者固有ライブラリルーチン208にメッセージを返し、供給業者固有ライブラリルーチン208はステータスをブロック706に返す。このステータスを受け取った後、ブロック708でこのライセンスが有効であるかどうか判定する。このライセンスが有効である場合、制御はブロック702に戻し、ライセンステーブル中の次のエントリをチェックする。このライセンスが有効でない場合は、制御はブロック708からブロック710に移り、ブロック710ではこのアプリケーションが無効なライセンスが検出されたときに呼び出すべきシグナルをセットアップしていただくか判定される。ライセンスは、そのライセンスの有効期限が過ぎているとき、あるいは（ネットワークの故障等のために）ライセンスサーバとの連絡が取れなくなったときに期限切れとなる。ライセンスサーバと連絡が取れないとき、サーバはいずれかの時点で別のモードで動作する別のアプリケーションにこのライセンスを再発行するかもしれない。また、アプリケーションによってはシグナルを取り扱えないものもあり、従ってかかるアプリケーションはLMS I ライブラリルーチン204を呼び出す際にシグナル機能をディスエールすることができる。その代わりに、このアプリケーションは図4に関連して上述したようにライセンスの現在の状態をシステムにたいして要求することができ、またライセンスの現在の状態をチェックするための呼び出しを要求することができる。

【0026】シグナルがセットアップされていない場合、ブロック710はブロック702に戻り、次のテーブルエントリをチェックする。シグナルがセットアップされている場合、制御はブロック710からブロック712に移り、ブロック712でライセンスが期限切れとなったことを知らせるシグナルをライセンスされたアプリケーションプログラム202に発する。

【0027】図8はライセンスモニタ変換テーブル（図2）の例を示す。図8において、ブロック802はNetLS供給業者固有ライブラリルーチンに対する変換テーブルエントリの例を示す。ブロック804はFlex 1m供給業者固有ライブラリルーチンに対する変換テーブルエントリを示し、ブロック806はSunNET供給業者固有ライブラリルーチンに対する変換テーブルエントリを示す。各ブロックに示すように、これらのエントリは位置に依存して意味が決まるものであり、そのためライセンスされたアプリケーションプログラムによって供給される情報は供給業者固有のライブラリルーチンの必要とする情報に簡単に変換することができる。当業者には、変換テーブルのセットアップには他の多くの方

法を用いることが理解されるであろう。

【0028】以上、本発明の現在の好適な実施例を説明したが、本発明の各側面が十分に達成されたことがわかるであろう。また、当業者には、本発明の精神と範囲から離れることなく本発明の構造、回路構成上の多くの変更、更に多岐にわたる実施態様及び応用を実施しうることが明らかであろう。この開示内容と説明は例示を目的とし、いかなる意味でも本発明を限定するものではない。本発明の範囲は特許請求の範囲によって理解されなければならない。

【0029】

【効果】以上詳細に説明したように、本発明によれば、アプリケーションから見るとライセンスサーバに対するインターフェースが統一されているので、アプリケーション中のライセンスサーバに関する部分の作成や保守が大層に簡単化される。また、ライセンスが得られなかった場合のユーザにたいする通知やそれにたいする処置のマンマシンインターフェースが、使用するライセンスサーバによらずに統一されたものになるので、使用しやすいシステムとなるという利点もある。

【図面の簡単な説明】

【図1】本発明が適用されたコンピュータシステムのブロック図。

【図2】本発明の一実施例のプロセスを示す概念図。

【図3】本発明の一実施例の初期化プロセスを示すフローチャート。

【図4】本発明の一実施例において、初期化後にアプリケーションプログラムから呼び出されてライセンスモニタと相互作用をする関数のフローチャート。

【図5】本発明の一実施例のライセンスモニタの一部分のフローチャート。

【図6】本発明の一実施例のライセンスモニタの一部分のフローチャート。

【図7】本発明の一実施例のライセンスモニタの一部分のフローチャート。

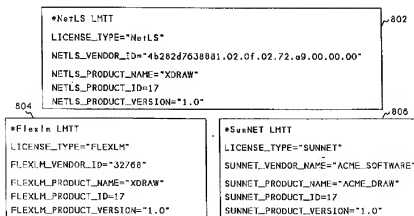
【図8】本発明の一実施例のライセンスモニタが使用する変換テーブルを示す図。

【図9】本発明の一実施例において、ライセンスが得られない場合にユーザの選択数を表示するユーザインターフェース画面を示す図。

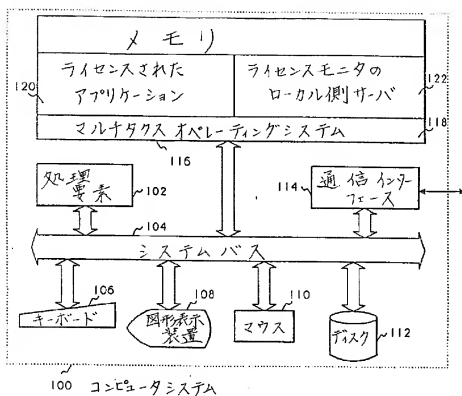
【符号の説明】

100: コンピュータシステム
102: 処理要素
104: システムバス
106: キーボード
108: 図形表示装置
110: マウス
112: ディスク
114: 通信インターフェース
116: メモリ
118: マルチタスクオペレーティングシステム
120: ライセンスされたアプリケーション
122: ライセンスモニタのローカル側サーバタスク
202: ライセンスされたアプリケーションプログラム
204: LMSI ライブラリルーチン
208: 供給業者固有ライブラリルーチン
210: ライセンスモニタ変換テーブル
212: LMSI ライセンスモニタ
214: 供給業者固有ライセンスサーバ
902、904、906、908: ボタン
910: ステータスボックス

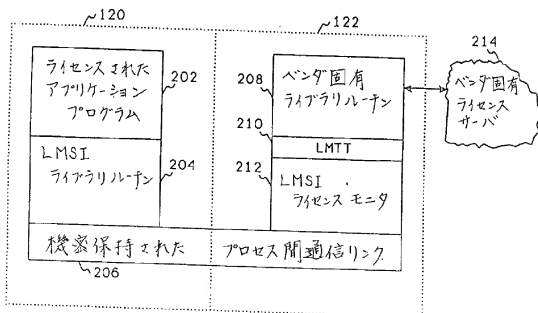
【図8】



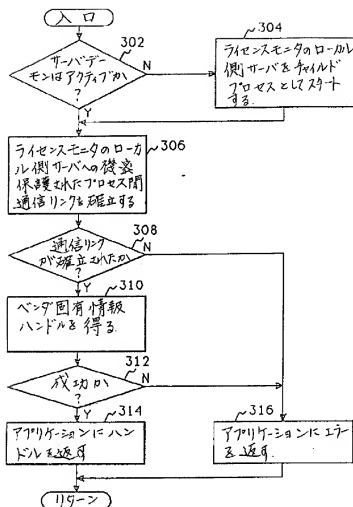
【図1】



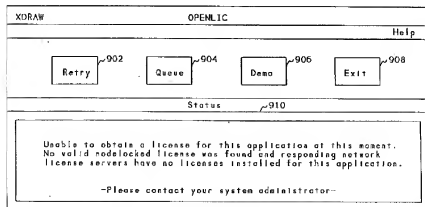
【図2】



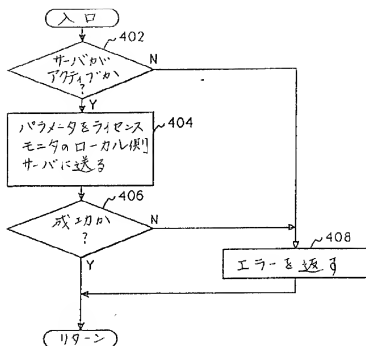
【図3】



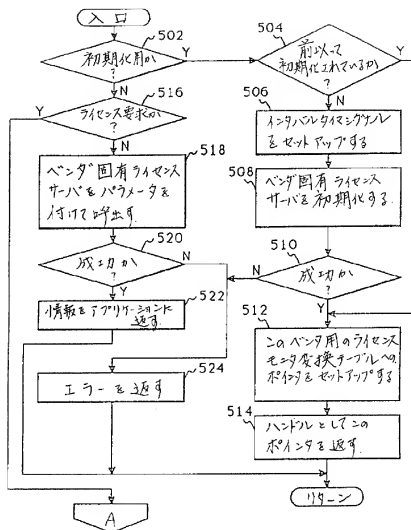
【図9】



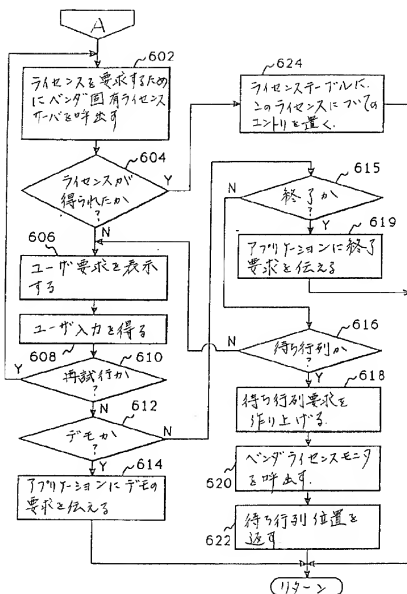
【図4】



【図5】



【図6】



【図7】

